Recommendation System for Social-aware Personal Processes

Special Thanks to:

Shawn Fang, Department of Computer Science, University of Texas at Austin
Dr. Anne Ngui, Department of Computer Science, Texas State University

ABSTRACT

The rise in popularity of various social network applications has brought the opportunities for Internet users to share and reuse a plethora of things like images, videos, datasets, maps, reviews etc. However, currently there is no effective way to share personal experiences such as the process of filling a personal income tax return or the process of applying a visa. We propose a social-aware personal process model and its implementation as a mobile application that empowers users to create, to execute and to share personal processes in the context of social network. As a social-aware personal process management system it is important to have an effective Recommendation System that predicts processes that a specific user may be interested aware personal process management system execute and to share personal processes in the context of social network. As a social-aware personal process management system it is important to have an effective Recommendation System that predicts processes that a specific user may be interested in using. This prediction is based on the similarity of users in their actions on existing processes. The assumption is: if two users have been copying and following the same processes then there is a decent probability that those two users have similar interests, which would be reflected as both of them will engage in similar personal processes in the future.

DATA AND RELATIONSHIP

- Each number (1 to 20) represents a Process ID, ex., Number “1” means Process ID = 101
- Each letter (A to E) represents a User ID, ex., Letter “A” means User ID = 101 in database
- Each line (Arrow, Curve, Straight) represents a relationship (Create, Copy, Follow, respectively) between a single User and a single Process. Ex., User A creates Processes 1 and 3, copies Processes 2, 4, 5, and 7, and also follows Process 6.

RECOMMENDATION ALGORITHM

- Target User = an input user who log in the PPM system
- Target Process = a process that Target User copied/created/follows
- Neighbor Category = a category that a target process belongs to
- Neighbor Category = a category that a neighbor process belongs to

Steps:
1. Target User
   1.1 Find a set of target processes and a set of target categories
   1.2 Find an average rating of each target category.

   Ex. Category 13 = (10+9.5+8)/3 = 9.167
   1.3 Find Jaccard Similarity between each neighbor user and target user and treat Jaccard Similarity as weighting.

   // Neighbor User: Voting
   1.4 Find a set of neighbor processes.
   1.5 Weighting all voting of a neighbor user by multiplying Jaccard Similarity. [Assume an action of copy or create a process as one vote and following a process as half vote]
   1.6 For each neighbor user, collect all weighted voting from every neighbor user who has either copied/created follows the process, and assign it as an Voting Score to the process.

   // Neighbor User: Rating
   1.7 A new process which only connects to its creator will never be recommended to other users unless people search for it. A content-based filtering could solve the problem. Since every process has a category when they are created and we have information of the target user has, that makes a connect between target user and the new process. Since there could be many new processes with same category, we can compare the content of target processes with same category to the new processes.

   2. Find a set of neighbor users
   2.1 Find a similarity between target user and the new process. Since there could be many new processes to the user, are possibly recommended to the user.

   3. Rank comprehensive scores from high to low
   3.1 Rank comprehensive scores from high to low
   3.2 For each neighbor process, collect all weighted voting from vary neighbor users who has the same category as the non-rated neighbor process.

   // Neighbor User: Rating
   3.3 If there is many voting to a particular process, the weight of rating score to comprehensive score could be regarded. Hence, a normalization to voting and ratings score could solve the problem.

   4. Normalize the comprehensive scores to the rating of this non-rated neighbor process. Otherwise, assign zero to it.

   // Comprehensive Score
   4.1 Sum the scores of Voting and Rating for each neighbor process
   4.2 Rank comprehensive scores from high to low

CONCLUSIONS

The recommendation algorithm brings a tree structure to the user of PPM. Once a user shares a common process with other users, all process relates to other users but exclude the known processes to the user, are possibly recommended to the user.

FUTURE WORK

The current recommendation system doesn’t take care of
1. If there is many voting to a particular process, the weight of rating score to comprehensive score could be regarded. Hence, a normalization to voting and ratings score could solve the problem.
2. If there is many voting to a particular process, the weight of rating score to comprehensive score could be regarded. Hence, a normalization to voting and ratings score could solve the problem.

REFERENCE


ACKNOWLEDGEMENT

We thank the National Science Foundation for funding the research under the Research Experiences for Undergraduates Program (CNS-1358939) at Texas State University to perform this piece of work and the infrastructure provided by a NSF-CRI 1305302 award.